Minimal Cooperation in Symport/Antiport P Systems with One Membrane

Artiom Alhazov[1,2], Yurii Rogozhin[2]

[1] Research Group on Mathematical Linguistics
Rovira i Virgili University
Pl. Imperial Tàrraco 1, 43005 Tarragona, Spain
E-mail: artiome.alhazov@estudiants.urv.es

[2] Institute of Mathematics and Computer Science
Academy of Science of Moldova
Str. Academiei 5, Chişinău, MD 2028, Moldova
E-mail: {artiom, rogozhin}@math.md

Summary. In this paper we consider symport/antiport P systems with one membrane and rules having at most two objects. Although it has been proved that only finite number sets can be generated by both $OP_1(sym_2)$ (one-membrane systems with symport rules of weight at most 2) and $OP_1(sym_1, anti_1)$ (one-membrane systems with symport/antiport rules of weight 1), the exact characterization is still an open question. We give some lower bounds, consider a few extensions, and state some open questions.

1 Introduction

Membrane systems (also called P systems) with symbol-objects are biologically motivated models of parallel distributed multiset processing – see details in [6]. Distributivity means that the objects (elements of a finite set) are placed in the compartments of the system, defined by a tree-like membrane structure. In symport/antiport P systems [5] – from now on we call them *communicative P systems* – the objects simply move across the membrane, alone or in groups, in the same direction (symport) or in the opposite directions (antiport). This feature is so powerful, that some restricted classes of these systems are already computationally complete.

Each object carries a finite amount of information. It is impossible to perform any nontrivial computation without interaction of different objects. This note is devoted to the study of the "weakest" communicative membrane systems: having at most one membrane and rules of transport of at most two objects.

2 Definitions

From now on by a number we mean a nonnegative integer. Let $NFIN$ represent the family of all finite number sets. We will denote by NRE the family of all recursively enumerable number sets, and by N_kRE, $k \geq 1$, the family of all recursively enumerable sets of numbers greater then or equal to k.

A communicative P system with m membranes is defined as a tuple

$$\Pi = (O, E, \mu, w_1, \cdots, w_m, R_1, \cdots, R_m, i_0),$$

where O is a finite set of objects (an alphabet), $E \subseteq O$ is a subset of objects present in the environment in unbounded quantities, μ is a hierarchical structure of m membranes, delimiting the regions, i_0 is the label of the output membrane, w_i is the initial multiset of objects in region i, $1 \leq i \leq m$, R_i is the set of rules associated to membrane i, $1 \leq i \leq m$.

The multisets are typically represented by strings with a corresponding multiplicity of each object. The rules can be of the following forms: (x, in) and (y, out) are symport rules, and $(y, out; x, in)$ are antiport rules, $x, y \in O^+$. Application of one of such rules (associated to some membrane i) consists of moving objects through membrane i: a multiset defined by x is moved from region i (contained in membrane i) into the immediately outer region and/or a multiset defined by y is moved from the region containing membrane i into the region i. The weight of a rule is the maximal number of objects moved in any direction ($|x|$, $|y|$ in the case of symport rules, and $max\{|x|, |y|\}$ in the case of antiport rules).

The rules are applied in a maximally parallel manner, non-deterministically. A configuration is called halting if no rules are applicable; a sequence of transitions among configurations is called a computation. The result of a halting computation is the number of objects present in the output membrane when the system halts. Collecting the results of all computations of Π, one obtains a number set generated by Π, denoted by $N(\Pi)$. By $NOP_m(sym_i, anti_j)$ we will denote the class of all number sets generated by P systems with at most m membranes, having symport rules of weight at most i and antiport rules of weight at most j.

3 Existing Universality Results

The results proved in [4]: $NOP_1(sym_3) \supseteq N_{13}RE$ and $NOP_1(sym_1, anti_2) \supseteq N_1RE$ (the latter result does not use rules exchanging 2 objects for 2 objects) show that the power of cooperation of 3 objects is enough for universality of P systems with one membrane. From now on, we will only consider the rules moving at most 2 objects.

It was proved in [2] that $NOP_3(sym_2) = NOP_3(sym_1, anti_1) = NRE$, so the power of cooperation of 2 objects is sufficient for the computational completeness of P systems with 3 membranes. The results in [3], that $DN_aOtP_2(sym_2) =$

$DN_aOtP_2(sym_1, anti_1) = NRE$ show that 2 membranes are enough for the tissue case (even for the deterministic computational completeness). Finally, it has been shown in [1] that $NOP_2(sym_2)_T = NOP_2(sym_1, anti_1)_T = NRE$: 2 membranes are enough to generate any recursively enumerable number set modulo the terminal alphabet (i.e., ignoring some objects in the output region).

4 Upper Bound

Mainly for these reasons we are especially interested in the following classes: $NOP_1(sym_1, anti_1)$, $NOP_1(sym_2)$, $NOP_1(sym_2, anti_1)$.

It has been shown in [3] that $NOP_1(sym_1, anti_1) \subseteq NFIN$. It has been shown in [4] that $NOP_1(sym_2) \subseteq NFIN$. The idea behind the proofs is that if a P system has a mechanism of continuously increasing the number of objects in region 1, then it cannot halt. These results are totally opposite to the completeness ones. We do not know whether $NOP_1(sym_2, anti_1) \subseteq NFIN$, but we conjecture that this is true.

5 Lower Bounds

At any step of the computation, each copy of an object can be in one of the two possible regions: in region 1 or in the environment. This is a kind of "1-bit memory", and the only way this memory can influence the following computation is by a cooperative transport rule: moving this object to the other region together with moving another object in the same or opposite direction. Therefore, it is expected that the number set generated by such P systems is "continuous", i.e., the distance between any two neighboring numbers is bounded.

More formally, let us use the following notion: the *segments* (finite segments of arithmetic progression with difference k) are defined as $SEG_k = \{\{n + ki \mid 0 \leq i \leq m\} \mid n, m \geq 0\} \cup \{\emptyset\}$. For instance, SEG_1 is the class of all sets of consecutive numbers, while SEG_2 is the class of all sets of consecutive even numbers and all sets of consecutive odd numbers.

Example 1. $\emptyset \in NOP_1(sym_1, anti_1) \cap NOP_1(sym_2)$.

Consider a P system $\Pi_0 = (O = \{b\}, E = \emptyset, \mu = [_1\ ]_1, w_1 = b, R = \{(b, in), (b, out)\}, i_0 = 1)$. There is one possible computation: object b oscillates between region 1 and the environment, so the set of results of the halting computations is empty. This system only uses symport rules of weight 1, so it belongs to both $OP_1(sym_1, anti_1)$ and $OP_1(sym_2)$.

Example 2. $NOP_1(sym_1, anti_1) \supseteq SEG_1$.

Fix the numbers $m, n \geq 0$. Consider a P system $\Pi_1 = (O = \{a, b\}, E = \{a\}, \mu = [_1\ ]_1, w_1 = a^n b^m, R = \{(b, out), (b, out; a, in)\}, i_0 = 1)$. Any computation of Π_1 halts in at most one step: every object b exit region 1, in exchange for either an object a or for nothing.

This is why the computation halts, with region 1 containing n copies of object a initially present there, and some number i of copies of a that were brought inside. Notice that $0 \leq i \leq m$, and every number is possible. Thus, $N(\Pi_1) = \{n + i \mid 0 \leq i \leq m\}$. Since m, n were chosen arbitrary, together with the previous example we obtain the result we claim: SEG_1 can be generated.

Example 3. $NOP_1(sym_2) \supseteq SEG_1 \cup SEG_2$.

Fix the numbers $m, n \geq 0$. Consider P systems $\Pi_2 = (O = \{a, b\}, E = \emptyset, \mu = [_1\ ]_1, w_1 = a^{n+m} b^m, R = \{(b, out), (ab, out)\}, i_0 = 1)$, $\Pi_3 = (O = \{a, b\}, E = \emptyset, \mu = [_1\ ]_1, w_1 = a^{n+2m} b^{2m}, R = \{(bb, out), (ab, out)\}, i_0 = 1)$. Any computation of Π_2 halts in at most one step: every object b exit region 1, together with either an object a or for nothing.

This is why the computation halts, with region 1 containing $n + m$ copies of object a initially present there, except some number j of copies of a that were taken outside. Notice that $0 \leq j \leq m$, and every number is possible. Substituting $j = m - i$, $0 \leq i \leq m$, we obtain $(n + m) - (m - i) = n + i$. Thus, $N(\Pi_2) = \{n + i \mid 0 \leq i \leq m\}$.

System Π_3 has a similar behavior, except there are $2m$ objects b initially present in region 1, and some number $2i$ of them leave region 1 in pairs, $0 \leq i \leq m$, while each of the others comes into the environment together with an object a. The number of objects a remaining in the system is $(n + 2m) - (2m - 2i) = 2i$. Therefore, $N(\Pi_3) = \{n + 2i \mid 0 \leq i \leq m\}$.

In this way, $SEG_1 \cup SEG_2$ can be generated by systems Π_2 and Π_3 for all possible numbers m, n, together with Π_0.

6 Extensions

It might be interesting to distinguish between the different objects obtained in region 1 when the system halts, thus obtaining vectors instead of numbers (in the notations, N is replaced by Ps). It might be also interesting to allow the environment to initially contain also some objects in finite multiplicities (*extended* environment; let us denote this feature with *eenv*).

One of the common features of segments introduced above is that of "continuity": there exists a number k such that the graph defined by numbers of the set as nodes, where two nodes i and j are adjacent if $|i - j| \leq q$, is connected. Let us define the distance between two vectors as a sum of absolute values of differences of their components. Then the notion of continuity can be extended to vector sets in the natural way: a vector set M is *k-continuous* if any two vectors in M belong to some sequence of vectors in M, such that the neighboring vectors of this sequence have distance at most k. Choose a number n. Below is an example of a P system with one membrane and transport rules of minimal cooperation that generates a vector set which is not n-continuous.

Example 4. (discontinuity for vectors)

Consider $\Pi_4 = (O = \{a, b, c, d\}, E = \emptyset, \mu = [_1\ \ ]_1, w_0 = b^n c^n d^{n+1}, w_1 = a^n, R, i_0 = 1)$, where $R = \{(a, out; b, in), (a, out; c, in), (a, out; d, in), (d, out; d, in), (bc, out), (bc, in)\}$ Consider a halting computation of Π_4. First, each object a must exit region 1, exchanged either for b or c (otherwise d will come inside the system and the computation will never halt). Then, the only way to make rules (bc, out), (bc, in) inapplicable is to separate object b and c, and this can only be done in the first step by the first two rules, and only if one of them is applied n times and the other one is not applied. Therefore, when Π_4 halts, region 1 will either contain b^n or c^n.

The distance between the two resulting vectors is $2n$. Notice that we took advantage of extended environment, and of both forms of minimal cooperation. We suppose that this result cannot be achieved with only one form of minimal cooperation, so we conjecture that $PsOP_1(sym_2, anti_1, eenv) \neq PsOP_1(sym_1, anti_1, eenv) \cup PsOP_1(sym_2, eenv)$.

7 Conclusions and Open Questions

The generative power of the first two classes of one-membrane communicative P systems with minimal cooperation considered above is between the class of finite number sets and the class of finite segments of arithmetic progressions (with difference 1 for $OP_1(sym_1, anti_1)$ and differences 1,2 for $OP_1(sym_2)$). It is still open what are the exact bounds, but we conjecture that $NOP_1(sym_1, anti_1) = SEG_1$ and $NOP_1(sym_2) = SEG_1 \cup SEG_2$ because, informally, the fact that the objects have "1-bit memory" can only influence one elementary decision.

It is interesting to consider the class of P systems with both forms of minimal cooperation: $OP_1(sym_2, anti_1)$. Does it generate more than $SEG_1 \cup SEG_2$? Does it only generate finite languages? We are unable to answer these questions, but we conjecture answer to the latter one is positive.

Finally, the similar questions were not studied much for the extensions, like allowing the environment to initially also have finite multiplicities of some objects. Does this extension increase the generative power of P systems with minimally cooperative communication and one membrane? We do not know the answer to these questions, but we conjecture that $NOP_1(sym_1, anti_1, eenv) \subseteq NFIN$ and $NOP_1(sym_2, eenv) \subseteq NFIN$ are true, i.e., while using only one form of cooperation, the generated number sets are still finite.

Acknowledgements

The first author is supported by the project TIC2002-04220-C03-02 of the Research Group on Mathematical Linguistics, Tarragona. Both authors acknowledge the Moldovan Research and Development Association (MRDA) and the U.S. Civilian Research and Development Foundation (CRDF), Award No. MM2-3034.

References

1. A. Alhazov, R. Freund, Yu. Rogozhin: Some optimal results on symport/antiport P systems with minimal cooperation. In *Cellular Computing. Complexity Aspects* (M.A. Gutiérrez-Naranjo, Gh. Păun, M.J. Pérez-Jiménez, eds.), Fénix Editora, Sevilla, 2005, 23–36.
2. A. Alhazov, M. Margenstern, Yu. Rogozhin, V. Rogozhin, S. Verlan: Communicative P systems with minimal cooperation. In *Membrane Computing. International Workshop* WMC 2004, Milan, Italy, 2004, Revised Selected and Invited Papers (G. Mauri, Gh. Păun, M.J. Pérez–Jiménez, G. Rozenberg, A. Salomaa, eds.), LNCS 3365, Springer-Verlag, Berlin, 2005, 161–177.
3. A. Alhazov, Yu. Rogozhin, S. Verlan: Symport/antiport tissue P systems with minimal cooperation. In *Cellular Computing. Complexity Aspects* (M.A. Gutiérrez-Naranjo, Gh. Păun, M.J. Pérez-Jiménez, eds.), Fénix Editora, Sevilla, 2005, 37–52.
4. P. Frisco, H.J. Hoogeboom: P systems with symport/antiport simulating counter automata. *Acta Informatica*, 41, 2-3 (2004), 145–170.
5. A. Păun, Gh. Păun: The power of communication: P systems with symport/antiport. *New Generation Computing*, 20, 3 (2002), 295–306.
6. Gh. Păun: *Membrane Computing. An Introduction.* Springer-Verlag, Berlin, 2002.